UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/545,040	04/07/2000	Gerard M. Col	CNTR:1568	9434

23669 7590 11/18/2004

HUFFMAN LAW GROUP, P.C.
1832 N. CASCADE AVE.
COLORADO SPRINGS, CO 80907-7449

EXAMINER	
LI, AIMEE J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 11/18/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/545,040	COL ET AL.	
	Examiner	Art Unit	
	Aimee J Li	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 May 2004 and 27 August 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-31,33 and 34* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-31,33 and 34* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-31 and 33-34 have been considered.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-6 rejected under 35 U.S.C. 103(a) as being taught by Popescu et al., U.S. Patent Number 5,487,156 (herein referred to as Popescu) in view of Preiss's "Direct vs. Indirect Containment" (herein referred to as Preiss).

4. Referring to claim 1, Popescu has taught an apparatus within a pipelined microprocessor for forwarding store instruction results to a pipeline stage for execution of a load instruction, the apparatus comprising:

a. A result forwarding cache (RFC), for storing at least one non-store instruction result destined for a user-visible register of the microprocessor (Popescu column 2, line 54 to column 3, line 9; column 6, lines 38-49; Figure 1; and Figure 5);

b. Comparison logic, for comparing a load address of the load instruction with a plurality of store addresses associated with said plurality of store instruction results to generate an address match signal (Popescu column 8, lines 29-36; Figure 1; and Figure 5); and

c. Control logic, configured to receive said match signal and selectively forward one of said plurality of store instruction results from said RFC to the pipeline stage in response to said match signal (Popescu column 1, lines 39-45; column 8, lines 35-59; Figure 1; and Figure 5).

5. Popescu has not taught storing a plurality of store instruction results. However, Popescu has taught storing the store instructions results's memory address, i.e. a pointer to the results (Popescu column 2, line 54 to column 3, line 9; column 6, lines 38-49; Figure 1; and Figure 5). Preiss has taught direct vs. indirect containment, i.e. storing the data directly or storing a pointer to the data (Preiss page 1). A person of ordinary skill in the art at the time the invention was made would have recognized that storing the results directly simplifies the cache's implementation. Also, if the result data is stored directly in the cache, there is less delay in retrieving the data, than if the system had to access the main memory location referred to by an indirect storage, i.e. the address, or pointer, to the results are stored. Please see the attached FOLDOC definition to see the inherent processor speed increase when only the cache needs to be accessed via direct storage rather than the cache then main memory via indirect storage. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the direct storage of store instruction results of Preiss in the device of Popescu to simplify the cache implementation and increase the speed of the processor.

6. Referring to claim 2, Popescu has not taught wherein said plurality of store instruction results comprise data to be stored from the microprocessor into a memory attached thereto. However, Popescu has taught storing the store instructions results's memory address, i.e. a pointer to the results (Popescu column 2, line 54 to column 3, line 9; column 6, lines 38-49;

Figure 1; and Figure 5). Preiss has taught direct vs. indirect containment, i.e. storing the data directly or storing a pointer to the data (Preiss page 1). A person of ordinary skill in the art at the time the invention was made would have recognized that storing the results directly simplifies the cache's implementation. Also, if the result data is stored directly in the cache, there is less delay in retrieving the data, than if the system had to access the main memory location referred to by an indirect storage, i.e. the address, or pointer, to the results are stored. Please see the attached FOLDOC definition to see the inherent processor speed increase when only the cache needs to be accessed via direct storage rather than the cache then main memory via indirect storage. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the direct storage of store instruction results of Preiss in the device of Popescu to simplify the cache implementation and increase the speed of the processor.

7. Referring to claim 3, Popescu has taught wherein said load address specifies a location of data to be loaded into the microprocessor from a memory attached thereto (Popescu column 8, lines 56-59 and Figure 5). In regards to Popescu, the address is inherent, since the address must be provided in order to return data from the specific address.

8. Referring to claim 4, Popescu has taught wherein said RFC comprises a plurality of storage elements for storing a predetermined number of instruction results (Popescu column 6, lines 38-49 and Figure 1).

9. Referring to claim 5, Popescu has taught wherein said instruction results are received by said RFC from an execution unit of the microprocessor (Popescu column 2, line 54 to column 3, line 9).

10. Referring to claim 6, Popescu has taught wherein said plurality of storage elements store said predetermined number of instruction results in a first-in-first-out manner (Popescu column 8, lines 29-59 and column 13, lines 67-65).

11. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Popescu et al., U.S. Patent Number 5,487,156 (herein referred to as Popescu) in view of Preiss's "Direct vs. Indirect Containment" (herein referred to as Preiss) and in further view of *In re Rose*, 220 F.2d 459, 105 USPQ 237 (CCPA 1955) (herein referred to as *In re Rose*). Popescu has not explicitly taught wherein said predetermined number of instruction results is five. However, Popescu has taught that the number of results that are stored varies (Popescu column 2, line 54 to column 3, line 9). A person of ordinary skill in the art at the time the invention was made would have recognized that the exact size of the buffer or cache does not matter since, no matter how many instruction results are present, the device functions the same. See *In re Rose*.

12. Claims 8-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Popescu et al., U.S. Patent Number 5,487,156 (herein referred to as Popescu) in view of Preiss's "Direct vs. Indirect Containment" (herein referred to as Preiss) and in further view of Abramson et al., U.S. Patent Number 5,606,670 (herein referred to as Abramson).

13. Referring to claims 8 and 9, Popescu has not taught

a. Wherein said load address and said plurality of store addresses comprise virtual addresses (Applicant's claim 8) (Abramson column 1, lines 36-52 and column 5, lines 8-13).

b. Wherein said virtual addresses comprise x86 linear addresses (Applicant's claim 9) (Abramson column 1, lines 36- 52 and column 5, lines 8-13).

14. However, Popescu has taught an out-of-order system with addresses (Popescu column 2, line 47 to column 3, line 9). Abramson has taught an out-of-order system (Abramson column 1, lines19-21)

a. Wherein said load address and said plurality of store addresses comprise virtual addresses (Applicant's claim 8) (Abramson column 1, lines 36-52 and column 5, lines 8-13).

b. Wherein said virtual addresses comprise x86 linear addresses (Applicant's claim 9) (Abramson column 1, lines 36- 52 and column 5, lines 8-13).

15. A person of ordinary skill in the art at the time the invention was made would have recognized that virtual addresses increase the number of address locations the system is able to reference, thereby increasing the address space. Therefore, it would have been obvious to a person of ordinary skill in the art at the time this invention was made to incorporate the virtual addresses of Abramson in the device of Popescu to increase the address space.

16. Claims 10-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Abramson et al., U.S. Patent Number 5,606,670 (herein referred to as Abramson) in view Popescu et al., U.S. Patent Number 5,487,156 (herein referred to as Popescu).

17. Referring to claim 10, Abramson has taught an apparatus for forwarding storehit data within stages of a pipelined microprocessor, the apparatus comprising:

a. A data unit, configured to forward a second plurality of store instruction results (Abramson column 2, lines 36-43; columns 4-5, lines 56-7; and column 6, lines 50-58); and

b. Selection logic, coupled to said RFC and said data unit, for selectively providing one of said first and second plurality of store instruction results to a stage of the microprocessor pipeline executing a load instruction (Abramson column 2, line 49 to column 3, lines 3).

18. Abramson has not taught a result forwarding cache (RFC), configured to forward at least one non-store instruction result, and to forward a first plurality of store instruction results. However, Abramson has taught that the system is an out-of-order system (Abramson column 1, lines 19-21). Popescu has taught an out-of-order system with a result forwarding cache (RFC), configured to forward at least one non-store instruction result, and to forward a first plurality of store instruction results (Popescu column 2, line 54 to column 3, line 9; column 6, lines 38-49; Figure 1; and Figure 5). A person of ordinary skill in the art at the time the invention was made, and as supported in Popescu, would have recognized that the RFC reduces stalls in a pipeline, thereby improving processor performance and efficiency (Popescu column 2, lines 47-53). Therefore, it would have been obvious to a person of ordinary skill in the art at the time this invention was made to incorporate the RFC of Popescu in the device of Abramson to increase processor performance and efficiency.

19. Referring to claim 11, Abramson has taught wherein said load instruction comprises a load address for specifying an address of data to be loaded into the microprocessor, wherein said selection logic is configured to forward one of said first and second plurality of store instruction results only if said load address matches one or more of a first and second plurality of store addresses corresponding to said first and second plurality of store instruction results (Abramson columns 2-3, lines 49-3 and columns 4-5, lines 56-7).

20. Referring to claim 12, Abramson has taught wherein selection logic forwards said first plurality of store instruction results forwarded by said RFC at a higher priority than said second plurality of store instructions results forwarded by said data unit if said load address matches both one or more of said first plurality of store addresses and one or more of said second plurality of store addresses (Abramson columns 2-3, lines 49-3).

21. Referring to claim 13, Abramson has taught further comprising comparison logic, coupled to said selection logic, for comparing said load address with said first and second plurality of store addresses to determine whether said load address matches one or more of said first and second plurality of store addresses (Abramson column 2, lines 49-66).

22. Referring to claim 14, Abramson has taught wherein said data unit is configured to forward said second plurality of store instruction results from a plurality of store buffers of the microprocessor (Abramson columns 5-6, lines 66-6 and column 6, lines 50-58).

23. Referring to claim 15, Abramson has taught wherein said plurality of store buffers is configured to store said second plurality of store instruction results while said second plurality of store instruction results are written to a memory coupled to the microprocessor (Abramson column 3, lines 53-56 and columns 4-5, lines 56-7).

24. Referring to claim 16, Abramson has taught wherein said data unit is configured to forward a newest one of said second plurality of store instruction results if said load address matches more than one of said second plurality of store addresses (Abramson column 1, lines 53-61).

25. Referring to claim 17, Abramson has taught wherein said RFC is configured to forward a newest one of said first plurality of store instruction results if said load address matches more than one of said first plurality of store addresses (Abramson column 1, lines 53-61).

26. Referring to claim 18, Abramson has taught an apparatus for detecting storehit conditions in a pipelined microprocessor in a hierarchical manner, the apparatus comprising:

a. First comparison logic, for comparing a load instruction load address in a first stage of the pipeline with a first plurality of store addresses of first store instruction data in a plurality of stages of the pipeline subsequent to said first pipeline stage (Abramson columns 2-3, lines 49-63);

b. Second comparison logic, for comparing said load address with a second plurality of store addresses of second store instruction data in a plurality of store buffers of the microprocessor (Abramson columns 2-3, lines 49-63); and

c. Control logic, coupled to said first and second comparison logic, configured to determine which of said first and second store instruction data is newest based on said first and second comparison logic comparing and to forward said newest store instruction data to said first pipeline stage in response thereto (Abramson columns 2-3, lines 49-3).

27. Abramson has not taught wherein said plurality of stages of the pipeline subsequent to said first pipeline stage do not comprise store buffers. However, Abramson has taught that the system is an out-of-order system (Abramson column 1, lines 19-21). Popescu has taught an out-of-order system wherein said plurality of stages of the pipeline subsequent to said first pipeline stage do not comprise store buffers (Popescu column 2, line 54 to column 3, line 9; column 6,

lines 38-49; Figure 1; and Figure 5). In regards to Popescu, the instruction shelf compares addresses of both store and non-store instructions. A person of ordinary skill in the art at the time the invention was made, and as supported in Popescu, would have recognized that the instruction shelves reduce stalls in a pipeline, thereby improving processor performance and efficiency (Popescu column 2, lines 47-53). Therefore, it would have been obvious to a person of ordinary skill in the art at the time this invention was made to incorporate the instruction shelving of Popescu in the device of Abramson to increase processor performance and efficiency.

28. Referring to claim 19, Abramson has taught wherein said first comparison logic is configured to compare virtual addresses (Abramson column 1, lines 53-62 and column 3, lines 57-61).

29. Referring to claim 20, Abramson has taught wherein said second comparison logic is configured to compare physical addresses (Abramson column 1, lines 53-62 and column 5, lines 8-13).

30. Claims 26-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Abramson et al., U.S. Patent Number 5,606,670 (herein referred to as Abramson) in view Popescu et al., U.S. Patent Number 5,487,156 (herein referred to as Popescu) and in further view of Preiss's "Direct vs. Indirect Containment" (herein referred to as Preiss).

31. Referring to claim 26, Abramson has taught a method for forwarding storehit data in a microprocessor pipeline, the method comprising:

a. Detecting a storehit condition, wherein a load instruction in a stage of the pipeline specifies data generated by a previous store instruction, wherein said data is still present in the pipeline (Abramson columns 2-3, lines 36-3);

b. Determining whether said data is present in said result forwarding cache (Abramson column 2, lines 36-43 and columns 4-5, lines 56-7);

c. Selectively forwarding said data from said result forwarding cache to said stage if said data is in said result forwarding cache (Abramson column 2, lines 36-43 and columns 4-5, lines 56-7); and

d. Selectively forwarding said data from a data unit of the microprocessor to said stage if said data is not in said result forwarding cache (Abramson column 2, lines 36-43; columns 4-5, lines 56-7; and column 6, lines 33-49).

32. Abramson has not taught storing at least one store instruction result pointer and at least one non-store instruction result into a result forwarding cache of the microprocessor. However, Abramson has taught that the system is an out-of-order system (Abramson column 1, lines 19-21). Popescu has taught an out-of-order system with storing at least one non-store instruction result into a result forwarding cache of the microprocessor (Popescu column 2, line 54 to column 3, line 9; column 6, lines 38-49; Figure 1; and Figure 5). A person of ordinary skill in the art at the time the invention was made, and as supported in Popescu, would have recognized that the forwarding cache reduces stalls in a pipeline, thereby improving processor performance and efficiency (Popescu column 2, lines 47-53). Therefore, it would have been obvious to a person of ordinary skill in the art at the time this invention was made to incorporate the forwarding cache of Popescu in the device of Abramson to increase processor performance and efficiency.

33. Abramson in view of Popescu has not taught storing a plurality of store instruction results. However, Popescu has taught storing the store instructions results's memory address, i.e. a pointer to the results (Popescu column 2, line 54 to column 3, line 9; column 6, lines 38-49; Figure 1; and Figure 5). Preiss has taught direct vs. indirect containment, i.e. storing the data directly or storing a pointer to the data (Preiss page 1). A person of ordinary skill in the art at the time the invention was made would have recognized that storing the results directly simplifies the cache's implementation. Also, if the result data is stored directly in the cache, there is less delay in retrieving the data, than if the system had to access the main memory location referred to by an indirect storage, i.e. the address, or pointer, to the results are stored. Please see the attached FOLDOC definition to see the inherent processor speed increase when only the cache needs to be accessed via direct storage rather than the cache then main memory via indirect storage. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the direct storage of store instruction results of Preiss in the device of Abramson in view of Popescu to simplify the cache implementation and increase the speed of the processor.

34. Referring to claim 27, Abramson has taught storing results data of each store instruction executed by an execution unit of the microprocessor in said result forwarding cache (Abramson columns 4-5, lines 56-7).

35. Referring to claim 28, Abramson has taught wherein said detecting said storehit condition comprises:

a. Comparing an address of said data specified by said load instructiol1 with a plurality of store instruction result data addresses stored in the pipeline below said stage (Abramson column 2, lines 49-66); and

b. Determining said address matches one or more of said plurality of data addresses (Abramson columns 2-3, lines 49-3).

36. Referring to claim 29, Abramson has taught wherein said determining whether said data is present in said result forwarding cache comprises:

a. Comparing an address of said data specified by said load instruction with a plurality of store instruction result data addresses stored in a predetermined number of stages of the pipeline below said stage (Abramson columns 2-3, lines 49-3);

b. Wherein said predetermined number equals a number of result entries in said result forwarding cache (Abramson columns 2-3, lines 49-3; column 4, lines 22-30; and column 6, lines 59-62).

37. Claims 21-25, 30-31, and 33-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Abramson et al., U.S. Patent Number 5,606,670 (herein referred to as Abramson) in view of Popescu et al., U.S. Patent Number 5,487,156 (herein referred to as Popescu) and in further view of Patterson and Hennessy's Computer Architecture A Quantitative Approach Second Edition © 1996 (herein referred to as Hennessy).

38. Referring to claim 21, Abramson has taught an apparatus for speculatively forwarding storehit data in a microprocessor pipeline, the apparatus comprising:

a. A plurality of virtual address comparators, for comparing a virtual load address with a plurality of virtual store addresses to generate a virtual match signal (Abramson column 1, lines 53-62; columns 2-3, lines 49-63; column 3, lines 57-61; and columns 7-8, lines 49-6);

b. A plurality of physical address comparators, for comparing a physical load address translated from said virtual load address with a plurality of physical store addresses translated from said plurality of virtual store addresses to generate a physical match signal (Abramson column 1, lines 53-62; columns 2-3, lines 49-63; column 5, lines 8-13; and columns 7-8, lines 49-6); and

c. Control logic, for receiving said virtual and physical match signals and generating a stall signal for stalling the pipeline subsequent to said forwarding logic forwarding said storehit data if said physical match signal indicates a match between said physical load address and one of said plurality of physical store addresses but said virtual match signal indicates no match between said virtual load address and one of said plurality of virtual store addresses (Abramson column 1, lines 36-52; columns 2-3, lines 49-3; column 3, line 53 to column 4, line 10; and column 8, line 66 to column 9, line 43).

39. Abramson has not taught forwarding logic, coupled to receive said match signal for forwarding the storehit data in response to said match signal indicating no match between said load address and said plurality of store addresses. However, Abramson has taught that the system is an out-of-order system (Abramson column 1, lines 19-21). Popescu has taught an out-of-order system forwarding logic, coupled to receive said virtual match signal for forwarding the

storehit data in response to said virtual match signal indicating no match between said virtual load address and said plurality of virtual store addresses, prior to generation of said physical match signal (Popescu column 8, line 29 to column 9, line 14 and Figure 5). In regards to Popescu, the instruction shelf compares addresses of store and non-store instructions. A person of ordinary skill in the art at the time the invention was made, and as supported in Popescu, would have recognized that the instruction shelves reduce stalls in a pipeline, thereby improving processor performance and efficiency (Popescu column 2, lines 47-53). Therefore, it would have been obvious to a person of ordinary skill in the art at the time this invention was made to incorporate the forwarding of Popescu in the device of Abramson to increase processor performance and efficiency.

40. In addition, Abramson has not taught generating a stall signal for stalling the pipeline. Hennessy has taught generating a stall signal for stalling the pipeline (Hennessy Pages 139 and 153). A person of ordinary skill in the art at the time the invention was made would have recognized that incorporating the stall of Hennessy would preserve the correct execution pattern when accessing slower memory. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the stall of Hennessy in the device of Abramson to ensure correct execution patterns.

41. Referring to claim 22, Abramson has taught the apparatus of claim 21, further comprising a data unit, configured to forward correct data specified by the load address to replace previously forwarded storehit data (Abramson columns 2-3, lines 49-3). Abramson has not taught wherein said control logic is configured to deassert said stall signal after said data unit forwards said correct data. Hennessy has taught to deassert the stall signal (Hennessy Pages 139 and 153). A

person of ordinary skill in the art at the time the invention was made would have recognized that incorporating the deassert signal of Hennessy would ensure the pipeline is not stalled longer than it needs to be stalled. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the deassert signal of Hennessy in the device of Abramson to increase speed.

42. Referring to claim 23, Abramson has taught a pipelined microprocessor for speculatively forwarding storehit data from a first pipeline stage to a second pipeline stage, wherein the storehit data is specified by a load address in the second stage, comprising:

a. Address region logic, configured to receive the load address and generate a match signal to indicate whether the load address is within one of a plurality of non-cacheable address regions of the microprocessor address space stored therein (Abramson columns 2-3, lines 49-3)

b. Forwarding logic, for forwarding the storehit data from the first stage to the second stage prior to said address region logic generating said match signal (Abramson columns 2-3, lines 49-3 and column 8, lines 43-49)

c. Control logic, configured to receive said match signal, subsequent to said forwarding logic forwarding the storehit data for provision to the second stage (Abramson columns 2-3, lines 49-3 and column 8, lines 43-49)

43. Abramson has not taught to assert a stall signal during a second clock cycle to stall the pipeline if said address region logic indicates the load address is within one of said plurality of non-cacheable address regions and to subsequently obtain using the load address non-cached correct data from a device external to the microprocessor. Hennessy has taught generating a stall

signal for stalling the pipeline (Hennessy Pages 139 and 153). A person of ordinary skill in the art at the time the invention was made would have recognized that incorporating the stall of Hennessy would preserve the correct execution pattern when accessing slower memory. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the stall of Hennessy in the device of Abramson to ensure correct execution patterns.

44. Referring to claim 24, Abramson has taught the microprocessor of claim 23, further comprising:

a. A bus interface unit, for receiving data from a bus coupled to the microprocessor, said bus further coupled to a system memory and a plurality of peripheral devices (Abramson column 1, lines 53-56 and column 4, lines 11-21)

b. At least one response buffer, operatively coupled to the second stage, for receiving load data specified by the load address from said bus interface unit, and for providing said load data to the second stage to replace the storehit data if the load address is within one of said plurality of non-cacheable address regions (Abramson column 1, lines 53-56; column 4, lines 11-21; and column 6, lines 33-58).

45. Referring to claim 25, Abramson has taught the microprocessor of claim 23, wherein said plurality of non-cacheable regions stored in said address region logic are software-programmable (Abramson column 6, lines 33-58). In regards to Abramson, it is inherent that the non-cacheable regions are software programmable since that is how data is written and changed.

46. Referring to claim 30, Abramson has taught a method for speculatively forwarding storehit data in a microprocessor pipeline, the method comprising:

a. Determining that a virtual load address matches no virtual store addresses present in the pipeline (Abramson column 9, lines 21-43);

b. Forwarding storehit data from a first stage to a second stage of the pipeline based on said determining that said virtual load address matches no virtual store addresses present in the pipeline (Abramson column 1, lines 53-62 and column 3, lines 57-1); and

47. Abramson has not taught detecting that a physical load address matches a physical store address present in the pipeline, subsequent to said forwarding said storehit data. However, Abramson has taught that the system is an out-of-order system (Abramson column 1, lines 19-21). Popescu has taught an out-of-order system detecting that a physical load address translated from said virtual load address matches a physical store address present in the pipeline, subsequent to said forwarding said storehit data (Popescu column 8, lines 29-36; Figure 1; and Figure 5). In regards to Popescu, the instruction shelf compares addresses of store and non-store instructions. A person of ordinary skill in the art at the time the invention was made, and as supported in Popescu, would have recognized that the instruction shelves reduce stalls in a pipeline, thereby improving processor performance and efficiency (Popescu column 2, lines 47-53). Therefore, it would have been obvious to a person of ordinary skill in the art at the time this invention was made to incorporate the instruction shelf of Popescu in the device of Abramson to increase processor performance and efficiency.

48. In addition, Abramson has not taught stalling the pipeline in response to said detecting. Hennessy has taught stalling the pipeline in response to said detecting (Hennessy Pages 139 and 153). A person of ordinary skill in the art at the time the invention was made would have recognized that incorporating the stall of Hennessy would preserve the correct execution pattern when accessing slower memory. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the stall of Hennessy in the device of Abramson to ensure correct execution patterns.

49. Referring to claim 31, Abramson has taught the method of claim 30, further comprising forwarding correction data from a third stage of the pipeline to said second stage after said stalling the pipeline (Abramson columns 2-3, lines 49-3). Abramson has not taught unstalling the pipeline after said forwarding said correction data. Hennessy has taught unstalling the pipeline after said forwarding said correction data. (Hennessy Pages 139 and 153). A person of ordinary skill in the art at the time the invention was made would have recognized that incorporating the unstall of Hennessy would ensure the pipeline it not stalled longer than it needs to be stalled. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the unstall of Hennessy in the device of Abramson to increase speed.

50. Referring to claim 33, Abramson has taught the method of claim 30, wherein said storehit data comprises a store instruction result within the pipeline having an identical physical store address as said physical load address (Abramson column 1, lines 36-39).

51. Referring to claim 34, Abramson has taught a method for speculatively forwarding storehit data in a microprocessor pipeline, the method comprising:

a. Detecting a storehit condition by comparing a load address with a plurality of store addresses (Abramson columns 2-3, lines 49-3)

b. Forwarding storehit data in response to said detecting said storehit condition (Abramson columns 2-3, lines 49-3 and columns 4-5, lines 56-7)

c. Determining said load address is within a non-cacheable address region subsequent to said speculatively forwarding (Abramson columns 2-3, lines 49-3 and column 6, lines 36-49)

52. Abramson has not taught stalling the pipeline in response to said determining said load address is within a non-cacheable address region. Hennessy has taught stalling the pipeline in response to said determining said load address is within a non-cacheable address region (Hennessy Pages 139 and 153). A person of ordinary skill in the art at the time the invention was made would have recognized that incorporating the stall of Hennessy would preserve the correct execution pattern when accessing slower memory. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the stall of Hennessy in the device of Abramson to ensure correct execution patterns.

Response to Arguments

53. Applicant's arguments filed 13 May 2004, with respect to the rejection(s)of claim(s) 1-9 and 26-29 under 102(b) and 103(a) respectively have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of the above.

54. Applicant's arguments filed 13 May 2004 have been fully considered but they are not persuasive.

55. Applicant's argue in essence on page 13 "With respect to claim 10...Applicant respectfully asserts that Popescu does not teach an RFC configured to forward a first plurality of store instruction results for the reasons stated above with respect to claim 1." This has not been found persuasive. Claim 1 states that the RFC stores the instruction results while claim 10 only states that the results are forwarded. There is nothing the claim to distinguish forwarding the results directly or indirectly as in claim 1 where there is a clear distinction that the results are stored directly. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., directly forwarding the results) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

56. Applicant argues on page 14, in essence

> Applicant respectfully asserts that the Examiner has failed to distinctly point out a stage in Popescu's processor that has a load instruction load address. Applicant respectfully asserts that the Examiner has also failed to distinctly point out a plurality of stages in Popescu's processor subsequent to the stage with a load address that have store addresses and that do not have store buffers.

57. This has not been found persuasive. Abramson was relied upon regarding the load addresses and load instructions. Popescu was relied upon to teach pipeline stages that have non-store buffers. Please see the rejection above for the citations related to each aspect of the claim. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on

combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

58. Applicant has argues on page 14, "The Examiner appears to rely on the fact that the processor taught in the two references are out-of-order processor as a motivation to combine the elements thereof." This has not been found persuasive. The motivation to combine was not that they are both out-of-order processors, but that the instruction shelving technique, which incorporates aspects claimed, reduces stalls and improves processor performance, as stated in the rejection above and taught in Popescu. Please see the above rejection for exact column and line citations. The statements that Popescu and Abramson are both out-of-order processors was just to demonstrate the similarity in the environments of Popescu and Abramson.

59. Applicant has argued on page 15 in essence

> Applicant respectfully asserts Popescu does not teach forwarding logic for forwarding storehit data, as described with respect to claim 1. Furthermore, Applicant can find no distinction in Popescu regarding virtual and physical addresses of load or store data, nor regarding comparison of same.

60. This has not been found persuasive. Claim 1 states that the RFC stores the instruction results while claim 21 only states that the results are forwarded. There is nothing the claim to distinguish forwarding the results directly or indirectly as in claim 1 where there is a clear distinction that the results are stored directly. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., directly forwarding the results) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the

specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Also, Popescu was not relied upon to teach virtual and physical addressing. Abramson was, and it was the combination of Abramson and Popescu that was relied upon to reject the claimed invention. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

61. Applicant argues in essence on page 16

> However, the invention of claim 23 does not even detect the hazard until it has already speculatively forwarded the storehit data to the load instruction from within the processor – without waiting to determine whether a hazard exists, and upon detection of the hazard, the invention subsequently stalls the pipeline for the purpose of correcting the violation of the now-detected hazard.

62. This has not been found persuasive. The speculative forwarded of storehit data was taught by Abramson, as cited in the rejection above. Hennessy was relied upon to merely show that stalling the pipeline after a hazard has been detected is common practice. Please see the rejection above to exact citations. It was the combination of Abramson, Popescu, and Hennessy that rejected the claimed invention. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

63. As a note, the arguments throughout the response seemed focused on one reference not teaching an entirety of the limitation(s) of the claims when the rejection was a combination of two prior art references. Applicant is reminded that in arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

Conclusion

64. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J Li whose telephone number is (571) 272-4169. The examiner can normally be reached on M-T 7:30am-5:00pm.

65. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

66. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AJL
Aimee J. Li
15 November 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100